Classification of Skin Cancer Images with Deep Learning

Group 5
Yung-Cheng (Allan) Chuang (s47451410) · Declan Fletcher (s46982494) · Jack Runchel (s47442454)
Ewan Stanich (s47428421) · Andrea Luca Taverna (s49321599)

1 Introduction

Over the past decade, deep convolutional neural networks (CNNs) (Krizhevsky, Sutskever, and Hinton, 2012) have established themselves as the standard methodology for image classification tasks, achieving human-level performance on benchmarks such as ImageNet. More recently, vision transformers (ViTs) (Chen et al., 2021; Dosovitskiy et al., 2020) have emerged as a powerful alternative, achieving competitive results on a variety of vision benchmarks. When properly tuned and combined with appropriate data augmentation and loss functions, these architectures have demonstrated remarkable accuracy and robustness across domains ranging from autonomous driving to fine-grained object recognition. In this report, we choose to investigate how such models perform in the medical field, focusing specifically on the diagnosis of dermoscopic images, which are magnified, polarized photographs of skin lesions that dermatologists use to aid in clinical decision-making.

We will explore our chosen models on the BCN20000 dataset (Hernández-Pérez et al., 2024), a collection of dermoscopic images curated expressly for machine learning research.

Accurate and meaningful classification of skin lesions is critical for early detection of malignant diseases (e.g., melanoma, basal cell carcinoma) and for determining appropriate treatment plans. Despite the importance of these diagnoses, manual interpretation remains time-consuming and heavily dependent on the experience of the clinician. Furthermore, even expert dermatologists achieve around 80–85% accuracy on average when discriminating between benign and malignant lesions (Brinker et al., 2021). This issue only deepens in regions with limited specialist availability and proficiency, emphasizing the importance of implementing such technology as an aid to these professions.

2 Related Works

Medical imaging has benefited immensely from deep learning. Esteva et al., 2017 have applied Inception-based models to classify over 120,000 clinical and dermoscopic skin-lesion images at dermatologist-level accuracy. ResNet50 has also proven highly effective: Pan et al., 2023 achieved 93.81% accuracy on retinal fundus scans and matched that performance on skin lesions. More recently, Vision Transformers have shown great promise: Park et al., 2022 employed a ViT to diagnose COVID-19 from chest X-rays with strong accuracy, highlighting the potential of both CNNs and ViTs for medical image classification.

3 Methods

We chose to investigate two pre-trained CNN architectures, Inception and Residual (ResNet) networks, as well as a pre-trained ViT model.

3.1 Inception Network

The Inception architecture was first introduced by Google researchers in 2014 under the name GoogLeNet (Szegedy et al., 2014). It was designed to strike a better balance between computational efficiency and model accuracy, particularly for large-scale image classification tasks. A later version, Inception v3 (Szegedy et al., 2016), introduced several improvements and has since become widely used due to its strong performance and relatively low resource requirements.

A key innovation in Inception is the use of parallel convolutional paths within a layer, typically applying 1×1, 3×3, and 5×5 filters along with pooling operations. These outputs are then concatenated, allowing the model to capture patterns at multiple spatial scales simultaneously. This design helps the model learn a more diverse set of features without a major increase in computation.

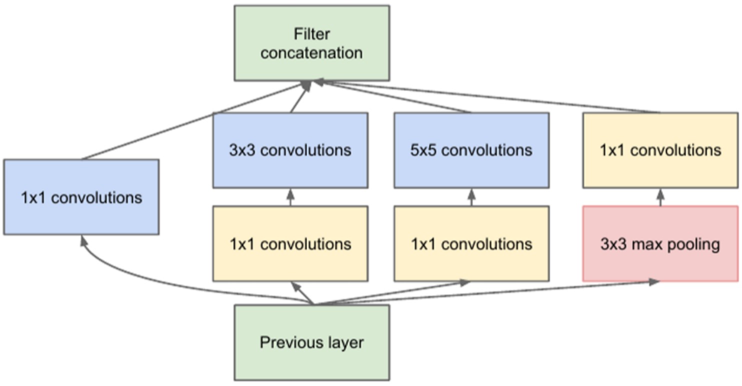


Figure 1: Inception Module with dimension reductions (Szegedy et al., 2014)

To reduce computational cost, 1×1 convolutions are used to compress the number of channels before applying larger filters. Inception v3 extends this by factorizing convolutions (e.g., replacing a 5×5 with two 3×3 operations) and using asymmetric filters like 1×3 followed by 3×1.

Inception v3 also includes auxiliary classifiers: additional branches that produce intermediate predictions during training. These help guide the learning process in early layers and improve gradient flow, reducing the risk of training instability.

Overall, Inception v3 delivers strong classification performance with far fewer parameters than deeper models. This makes it particularly well-suited for applications such as medical imaging, where computational efficiency is important but high accuracy remains critical.

In our project, we used the pretrained Inception v3 model provided by the PyTorch library (*Inception v3 - PyTorch Hub* 2019), which was originally trained on the ImageNet dataset.

3.2 Residual Network

The ResNet family of models, first proposed by He et al. (He et al., 2015), remains the gold standard for ImageNet-scale classification. While traditional ResNets increase representational capacity by adding more layers, the Wide ResNet variant (Zagoruyko and Komodakis, 2016) instead increases the width (number of channels) of each convolutional layer to boost accuracy and training stability. For our experiments, we adopted PyTorch's Wide ResNet-101, making use of pretrained weights from training on ImageNet (Torchvision Contributors, 2025). Figures 2 and 3 illustrate its high-level structure and the bottleneck residual block, respectively.

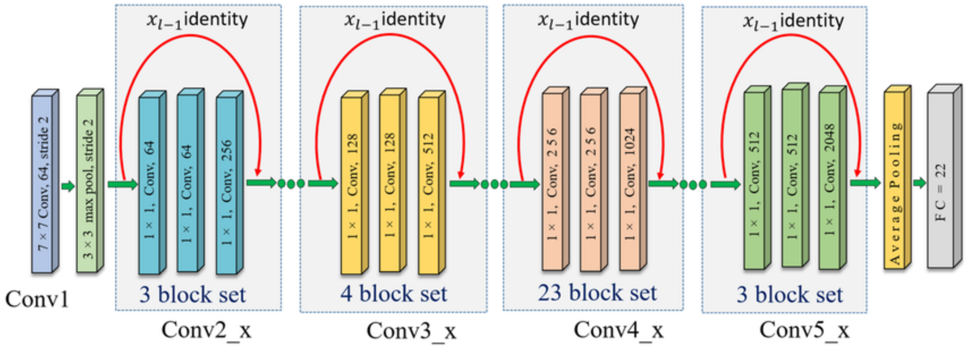


Figure 2: The architecture of Pytorch's Wide ResNet101, broken into specific segments that are repeated the specified times (Liu, 2023).

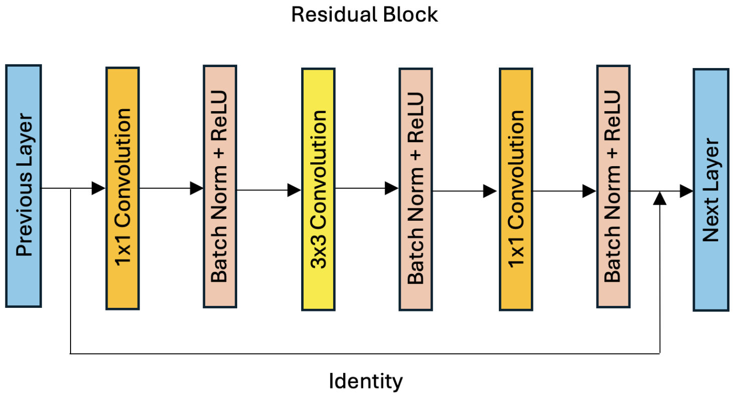


Figure 3: An individual residual block in Pytorch's Wide ResNet101.

This architecture involves four stages of repeated residual blocks, where the number of feature channels increases in a fixed pattern: 64 → 256, 256 → 512, 512 → 1024, and 1024 → 2048. Each block consists sequentially of a 1×1 convolution for dimensionality reduction, a 3×3 convolution for spatial processing, and a final 1×1 convolution to restore dimensionality. Batch normalisation and ReLU activation are applied after each convolution, following the post-activation design. Common with most ResNet models, the skip connection is typically an identity mapping, unless the input and output dimensions differ, in which case a 1×1 convolution is used to align them; this occurs when changing into a new stage of residual blocks.

To adopt this model to our problem, we applied a softmax function to the output head to ensure that the model was producing class probabilities for the eight classes of interest. A dropout of 0.3 was also introduced into the final fully connected layer to improve generalization through model averaging.

3.3 Vision Transformer

Transformers—first introduced for language translation (Vaswani et al., 2023)—revolutionized deep learning by capturing the context of each word based on surrounding words. Recently, the same attention mechanism was adapted for image data based on the idea that parts of images have different meanings based on their surroundings. Specifically, the Vision Transformer (ViT) was created by Dosovitskiy et al., 2021 to allow image data to be used with a Transformer. For this project, we used the ViTForImageClassification model from Hugging Face (Hugging Face, 2025), specifically the `google/vit-base-patch16-224-in21k` variant, which is pretrained on ImageNet.

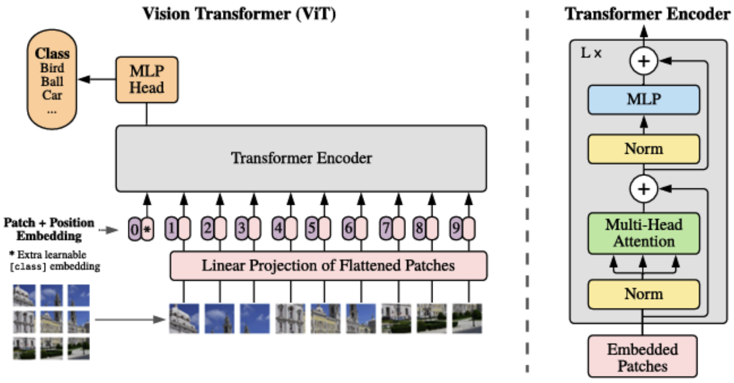


Figure 4: Overview of Visual Transformer from Dosovitskiy et al., 2021.

We now describe how the ViT can input images to a transformer encoder (see Dosovitskiy et al., 2021 and Vaswani et al., 2023 for more details). The ViT splits input images into $P \times P = 16 \times 16$ input patches. Each patch $\mathbf{x}_p^1, \ldots, \mathbf{x}_p^N \in \mathbb{R}^{P \times P \times C}$, where C is the number of channels, is then flattened to a single layer and mapped to $D = 768$ dimensions with a trainable linear projection $\mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}$. An additional learnable class token $\mathbf{x}_{\text{class}} \in \mathbb{R}^D$ is prepended to these patch embeddings; the output of this token following the Transformer encoder is used as the image representation and is passed to the MLP classifier head. Finally, a learnable position embedding $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N+1) \times D}$ is added to retain positional information (this is necessary since the Transformer encoder does not retain the 2D image structure like a CNN does). The final input to the Transformer encoder is given by

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1 \mathbf{E}; \cdots ; \mathbf{x}_p^N \mathbf{E}] + \mathbf{E}_{\text{pos}}.$$

This sequence of embedding vectors $\mathbf{z}_0$ is then passed to the Transformer encoder (shown on the right of Figure 4). This encoder consists of $L = 12$ layers, each consisting of a 12-head self-attention block and a 2-layer MLP block. Layer normalization is applied between every block and residual connections are made after every block. The output of the transformer encoder after the 12th layer is denoted $\mathbf{z}_{12} = [\mathbf{z}_{12}^0; \cdots ; \mathbf{z}_{12}^N]$, and as mentioned, the row $\mathbf{z}_{12}^0$ corresponding to the class token is fed as input to the MLP classifier head. During inference, this is a single linear layer.

4 Data

We used the BCN20000 dataset (Hernández-Pérez et al., 2024). This is a dataset of 18,946 dermoscopic images released for machine learning research (see Figure 5 for examples); each image has resolution 1024 × 1024, though they were downsampled to the appropriate size for each model during training and testing. The dataset specifically aims to address the problem of unconstrained classification of dermoscopic images by including lesions in regions which are hard to diagnose (such as under finger nails), and lesions which do not fit in the aperture of the dermoscopy device. The dataset was also used as part of the International Skin Imaging Collaboration (ISIC) machine learning competition ISIC-2019 (Wu et al., 2022).

The images and metadata for the dataset were downloaded to the training cluster using the ISIC Archive API (ISIC, 2025). Three diagnosis categories in the dataset are not true skin cancers: scar, melanoma metastasis and other. The “other” category corresponds to lesions

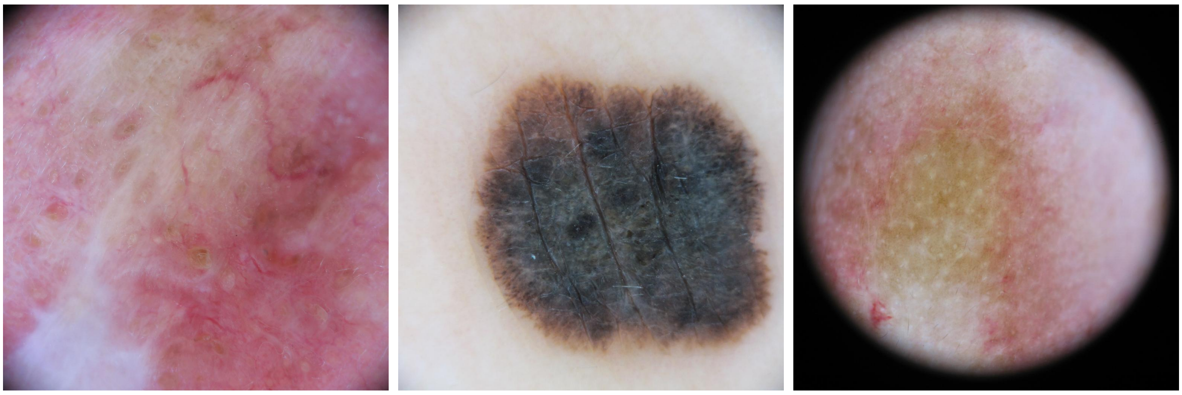


Figure 5: Examples of images in the BCN20000 dataset.

which do not fall into one of the eight main categories, and is considered to be an out-of-distribution class. We removed these three classes to focus on classifying skin cancer lesions, and ensure the models accuracy could be evaluated. The resulting dataset contained 16,843 images. The distribution of classes in the final dataset is shown in Table 3; note the classes are highly unbalanced.

The dataset was randomly split into train (70%), validation (20%) and test (10%) sets, stratified on the class to ensure each set had equal proportions of classes. This split was used for each model for comparability.

Diagnosis/class	**Type**	**Count**	**%**
Melanoma	Malignant	4003	23.8
Nevus	Benign	5647	33.5
Basal cell carcinoma	Malignant	3676	21.8
Squamous cell carcinoma	Malignant	559	3.3
Dermatofibroma	Benign	168	1.0
Benign keratosis	Benign	1551	9.2
Actinic keratosis	Malignant	1088	6.5
Vascular lesion	Benign	151	0.9
Total		**16843**	**100**

Table 1: Distribution of classes after removing images in unwanted classes.

5 Experiments

5.1 Training techniques

We used the cross entropy loss when training, as this is the standard choice for multi-class classification. All models were trained on A100 GPU nodes of the same training cluster. To train each model, we used the following procedure. First, we trained a "base" model directly on the dataset without any optimisation or regularisation techniques, to assess the feasibility of each model for our task. Then, we ran many experiments using different techniques to improve validation accuracy. These techniques include:

- Hyperparameter optimisation: carefully optimising learning rates, batch sizes, etc.

- Data augmentation: using randomly perturbed training examples to avoid overfitting to irregularities in the data set. (More details are given §5.2—§5.4 below, and the parameters of the transformations used are in Appendix A).
- Balanced class sampling: sampling training examples from each class with equal probability to aim for balanced performance across the unbalanced classes.
- Class-weighted loss: weighting the cross-entropy loss to penalise the model more for misclassifying smaller classes, again, to combat class imbalance.
- Adversarial data and class balancing: during ViT training, an adversarial data set was created. Specifically, after ViT Model 1 (see §5.4) was trained, this model was used to create adversarial examples using the FGSM algorithm with $\epsilon \sim U(0.01, 0, 1)$. Furthermore, the adversarial examples were created in proportions that balanced the number of samples in each classes (for the combined dataset of original and adversarial examples); for each class, images from the original dataset were randomly selected together with a random ϵ to create an adversarial example. This was repeated until every class had 5647 images in total (to match the original number of samples in the nevus class).[1]
- Attention-based regularization: ensure the ViT is not overfitting to specific patches using the attention mechanism (see §5.4 for more details).

We now explain how these techniques were used to train each architecture.

5.2 Inception

Our Inception models were all trained for 25 epochs using the PyTorch pretrained Inception v3 framework. Before each model was trained the images were resized to 299 × 299 to fit the input resolution for the framework and normalized as shown in Appendix A. A composite loss function was used for each model, defined as

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + 0.3\,\mathcal{L}_{\text{aux}}$$

where $\mathcal{L}_{\text{CE}}$ is the primary cross-entropy loss and $\mathcal{L}_{\text{aux}}$ is the auxiliary branch loss utilizing Inception's built-in regularisation.

Base model: We began investigating the Inception model with a simple implementation of the Inception v3 trained with a batch size of 32 and an Adam optimizer with a learning rate of 5×10^{-4} and weight decay of 1×10^{-4}. The images were resized and normalized, but no further transforms were applied. This model achieved 76% accuracy on the validation set although the loss plot (see Appendix B) shows unstable learning with significant divergences in both loss and accuracy. Some adjustments needed to be made to improve that stability of the learning and generalisation of the model.

Model 1: Building on the original model, we aimed to stabilise the training. We lowered the learning rate to 1×10^{-4} and implemented a learning rate scheduler decreasing the learning rate by a factor of 0.5 every 3 epochs to smooth the parameter updates and avoid erratic optimizer learning. A weight decay of 1×10^{-4} was included to penalise large weights and the batch size was also increased to 64 to reduce gradient noise. With these adjustments, the learning was significantly smoother with a more gradual decrease in training and validation loss achieving a validation accuracy of 82%, an improvement over the baseline model. Although the model stability and accuracy improved, the validation accuracy quickly plateaued and eventually marginally decreased, clear signs of over fitting and poor generalisation.

[1] We are using the FGSM method to create synthetic examples as a regularization technique, rather than specifically trying to defend against adversarial data.

Final model: To address the overfitting and significant class imbalances, some further improvements were made. Our improved Inception v3 model used the set of data augmentations listed in Appendix A. These augmentations were largely drawn from the dataset's original paper (Hernández-Pérez et al., 2024) where other CNN models were applied to the same dataset. We imposed some changes to the augmentations that ultimately improved the generalisation and classification performance for our Inception v3 CNN on the skin lesion dataset. We restricted the random resized crop scale and excluded both random vertical flip and grey-scaling a portion of the images. We also added a ±15° random rotation step and reduced the colour-jitter hue to 10% because the Inception v3 exhibited sensitivity to small orientation and colour changes. All other colour-jitter parameters and augmentations remained identical to the original paper.

Like the original paper, a weighted sampler was also implemented to ensure approximately uniform class distributions, which proved critical given our dataset's heavy class imbalance. Weighted cross-entropy loss was also considered, however, despite improvements on underrepresented classes' recall, decreased performance on the larger classes outweighed the improvements and was therefore omitted. Dropout was also considered in the model to improve generalisation by preventing overfitting, however, on the smaller Inception model, we found too much information was lost and was also not used in the final model. After experimenting with several optimizers (Adam, SGD, and RMSProp) and tuning hyperparameters, we found that an Adam optimizer with a learning rate of 1×10^{-4} and a weight decay of 1×10^{-4} provided the best results. This final model achieved 84.97% accuracy on the validation set.

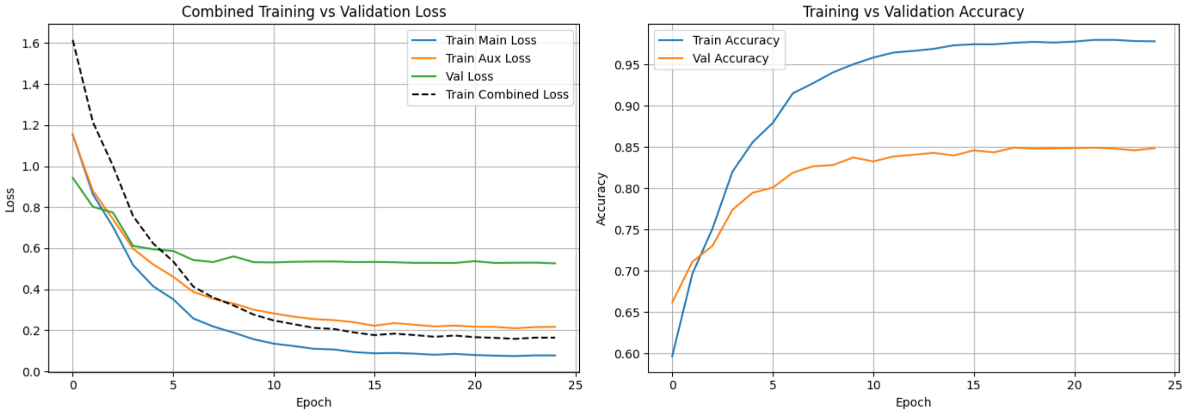


Figure 6: Inception v3 Training and Validation, Loss and Accuracy.

The final model took approximately 1 hour and 30 minutes to train and showed better generalisation and significantly smoother training after the improvements, as shown in Figure 6 above. The training and validation accuracies both plateau at about 98% and 85% respectively, suggesting overfitting is still an issue and the robustness could be further improved.

5.3 ResNet

All ResNet models were trained using PyTorch's Adam optimizer.

Base model: A baseline WideResNet101 model was trained for 20 epochs with a batch size of 64. Default parameters were used for the Adam optimizer, specifically a learning rate of 1×10^{-3} and no weight decay. The only preprocessing consisted of resizing the images and applying standard ImageNet normalization. This configuration yielded a validation accuracy of 76%. However, the training was turbulent, with noticeable fluctuations in the loss curve (see

Appendix B), suggesting unstable convergence and potential overfitting or sensitivity to the learning rate.

Model 1: To boost performance and enhance robustness, data augmentation was incorporated into the pipeline. This decision was motivated by the lack of diversity in the training data, which likely limited the model's ability to generalise to real-world conditions. Following the approach used in the Inception model, the transformations (outlined in Appendix A) were adapted from the dataset paper and enhanced through hyperparameter tuning and additional augmentations.

By exposing the model to varied representations of the same class, these augmentations helped the network become invariant to common transformations such as rotation, scaling, and lighting changes. This improved model achieved a validation accuracy of 80%.

Final model: Several more key adjustments were introduced in an effort to stabilise training and improve generalisation. The learning rate was reduced to 1.5×10^{-4} and a weight decay of 1.5×10^{-3} was added. These choices were driven by the observation of noisy loss behaviour in the base model. A lower learning rate was hypothesised to provide smoother, more stable updates, while weight decay was introduced to counter overfitting by penalising large weights.

We also experimented with the use of the adversarial dataset (described in §5.1) to mitigate class imbalance, similar to the approach used with the ViT (see §5.4). This did not yield any substantial improvement in performance, possibly due to ineffective hyperparameters or methods, as not all could be trialled due to time and computational constraints.

The batch size was reduced to 32 based on the hypothesis that smaller batches introduce more variability during training, which can help the model learn more general features and improve performance on unseen data. Additionally, a linear learning rate scheduler was introduced, with a step size of 7 epochs and a gamma value of 0.1. This scheduler allowed larger learning rate updates in the early stages of training, when the model is still far from an optimal solution, and smaller updates later on to fine-tune performance as training progresses. Together, these adjustments were aimed at achieving a more stable optimization trajectory and better generalisation. This final model achieved an accuracy of 85.86% on the validation set.

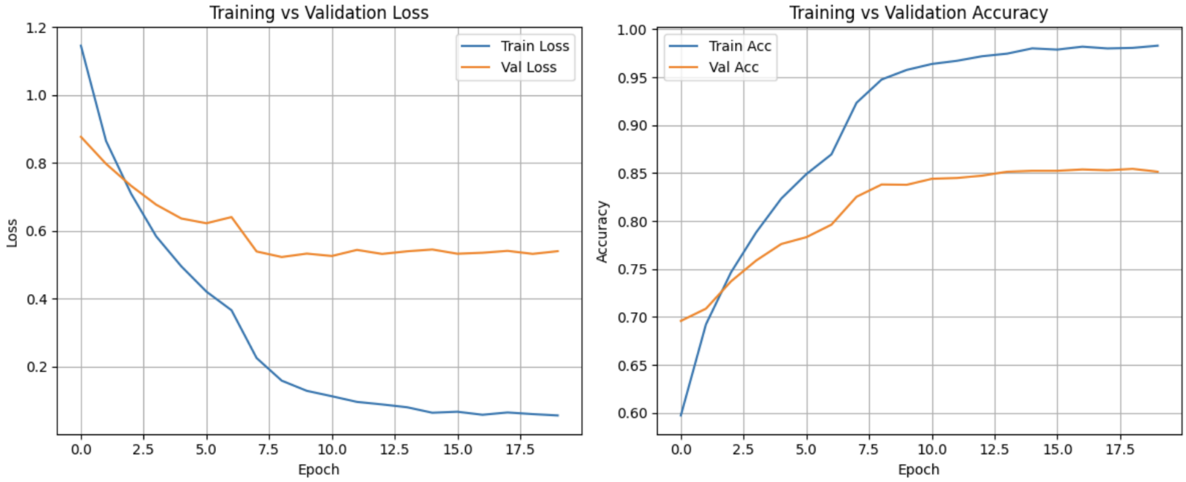


Figure 7: ResNet Training and Validation Loss and Accuracy.

It is interesting to note that, although both Inception and ResNet are deep CNNs, they did not respond equally to the same training techniques. Most notably, the Inception model performed better when trained with a weighted sampler, whereas the ResNet model experienced a significant drop in performance under the same condition. Additionally, the ResNet responded much better to dropout and a smaller batch size; there were also slight differences in the optimal learning rate and weight decay values for the Adam optimizer across the two architectures.

The final model took approximately 1 hour and 50 minutes to train. As shown in Figure 7, while the training accuracy continues to increase steadily, the validation accuracy plateaus around 85%, indicating that overfitting is still an issue. The continued gap between training and validation performance in the later epochs highlights this discrepancy. Although improvements such as weight decay and data augmentation helped, the model may still benefit from stronger regularisation.

5.4 ViT

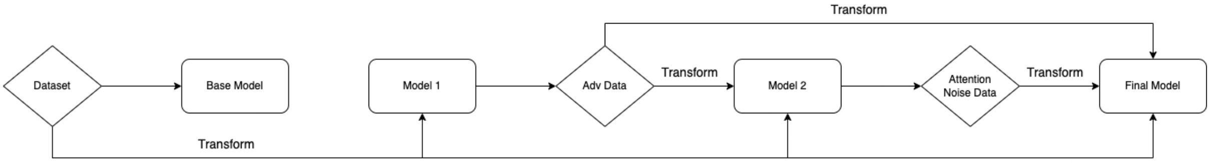


Figure 8: Overview of ViT improvement.

All ViT models were trained using PyTorch's Adam optimizer. Unless stated otherwise, the optimizer parameters were left to PyTorch defaults.

Base model: A base ViT was trained for 100 epochs using a learning rate of 2×10^{-6} and a batch size of 128. This resulted in a validation accuracy of 75%, though the loss curve (see Appendix B) suggested the model was slightly overfit. Furthermore, the classification report showed the model had poor performance on small classes.

Model 1: To reduce the overfitting seen in the base model, data augmentation was used for regularisation. Specifically, during training images were randomly transformed using the transformations listed in Appendix A. A learning rate of 1×10^{-5} and a batch size of 256 was used, and the model trained for 100 epochs. With this technique, the validation accuracy improved to 83%.

Model 2: Despite this improvement, class imbalance remained an issue. Attempts to resolve this using a class-weighted cross entropy loss function were unsuccessful. Improvements for the precision and recall of smaller classes were observed but at the expense of reduced accuracy for the large classes. A class-weighted loss was probably ineffective on its own due to the fact that there are two few training samples in the small classes for the model to learn them correctly.

Bagging and boosting methodologies were considered but deemed impractical for the large ViT model due to long training times and the initial model's near-100% training accuracy, which left no diverse data for subsequent boosting iterations.

However, further data augmentation techniques were considered. Specifically, Model 1 was used to generate adversarial examples as described in §5.1. This process ensured that with the synthetic examples, all eight classes had an equal count in the training set, effectively resolving the imbalance. The random transformations listed in Appendix A were again used during training on the balanced dataset. The model was trained for 100 epochs with a learning rate of 1×10^{-5} and a batch size of 256. This Model 2 achieved an impressive 86% validation accuracy.

Final model: Inspired by the concept of attention mechanisms, it was hypothesized the adversarial trained model might overfit on specific image patches. To mitigate this, a new dataset was created with examples that are perturbed in a particular way using the attention mechanism; we now explain the details.

Using Model 2, for each image, the attention of the last encoder layer was averaged over all 12 heads and then normalized to lie in $[0, 1]$. Specifically, if A denotes the tensor of mean attention values averaged over the 12 heads, we computed

$$\text{Patch importance} = \frac{A - \min(A)}{\max(A) - \min(A)}.$$

This metric was used to add noise to random images to ensure the new model would not overfit to specific patches. The perturbed images were created in the following way; for each image, there is a 60% chance of having noise added. Of those images, any patch with importance score greater than 0.6 had Gaussian noise with $\mu = 0$ and $\sigma^2 \sim U(20, 60)$ added to the RGB values.

The final model was trained on the attention-noise dataset, adversarial dataset, and original dataset. A learning rate of 8×10^{-6}, a batch size of 150 and 115 epochs were used for training. This final model took 20 hours to train and resulted in 88.69% accuracy on the validation set.

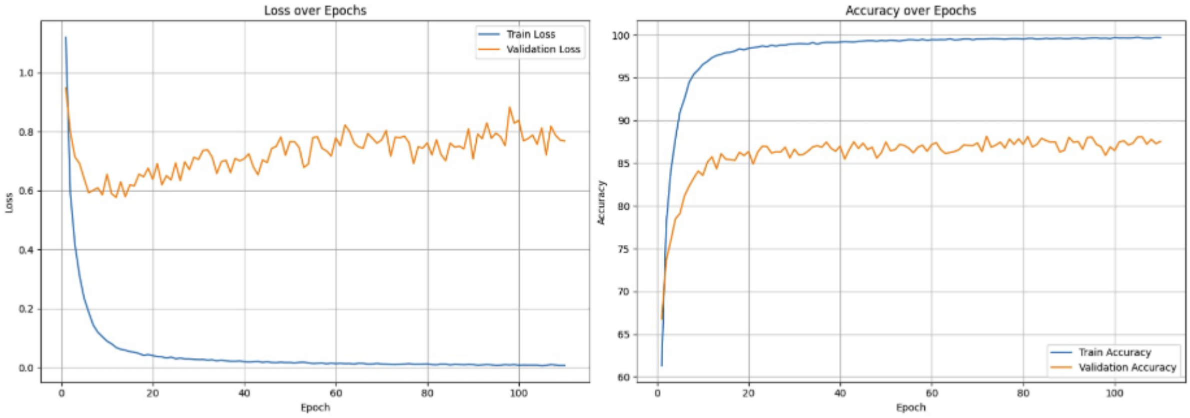


Figure 9: Final ViT model training and validation loss and accuracy.

While the model accuracy did improve, Figure 9 shows the model training is very unstable from a validation loss perspective, with it showing signs of overfitting after approximately 10 epochs. However, the validation accuracy (which is our primary focus) continues to increase by a small gradient. Despite the unstable training, we used this model for its strong classification accuracy. More stable training and better performance may be possible with further hyperparameter tuning.

6 Results

We now present the test set performance of the final models for each architecture.

6.1 Accuracy and Balanced Accuracy

We first measure the performance of the models using accuracy and balanced accuracy. The balanced accuracy is the average recall for each class; if there are C classes and the i^{th} class has TP_i true positives and FN_i false negatives, the balanced accuracy is

$$\frac{1}{C} \sum_{i=1}^{C} \frac{TP_i}{TP_i + FN_i}.$$

Balanced accuracy is often used to evaluate classification models with unbalanced classes.

Model	**Accuracy**	**Balanced Accuracy**
Inception	85.40	75.61
ResNet	88.19	79.27
ViT	**89.67**	**81.03**

Table 2: Test set accuracy and balanced accuracy (in %). Best result bolded.

Overall, all models were able to achieve strong classification accuracy above 85%. The ViT had the best accuracy and balanced accuracy, but the Inception and ResNet still performed very well. It is interesting to note all models had higher test accuracy than validation accuracy, suggesting the test set contained images which were easier to classify.

There are two important comparisons we can make with previous results in the literature. In Hernández-Pérez et al., 2024, baseline CNN models were fit to the BCN20000 dataset. The best model was an EfficientNet-B2, which achieved a balanced accuracy of 46.05%, so all our models have much stronger balanced accuracies than their best baseline. Also, Wu et al., 2022 notes that the winner of the ISIC-2019 competition achieved a balanced accuracy of 74.2% on the BCN20000 dataset, so we have achieved comparable or better performance with all models. Although these comparisons are not perfectly fair (most notably, our test set did not contain out-of-distribution samples whereas the ISIC-2019 competition dataset did), our models perform very well for the task.

6.2 Recall for malignant classes

Although overall accuracy provides an important overview of model performance, more detailed metrics for a multi-class classification problem can be considered. Specifically, precision and recall can be computed using a classification report (see Appendix C), and exact errors can be examined in a confusion matrix (see Appendix D). We will consider recall for key classes.

The recall of a class measures the proportion of true cases that were identified. Specifically, if the i^{th} class has TP_i true positives and FN_i false negatives, the recall for the class is

$$\frac{TP_i}{TP_i + FN_i}.$$

In a medical context, high recall for malignant classes is imperative. In particular, high recall means there will be few false negatives and reduced chance of missing life-threatening diagnoses.

Diagnosis	**Inception Recall**	**ResNet Recall**	**ViT Recall**	**Support**
Melanoma	87.00	87.75	**89.75**	400
Basal cell carcinoma	88.32	91.85	**93.48**	368
Squamous cell carcinoma	62.50	**76.79**	66.07	56
Actinic keratosis	76.15	73.39	**80.76**	109

Table 3: Test set recall for malignant classes. Best result bolded.

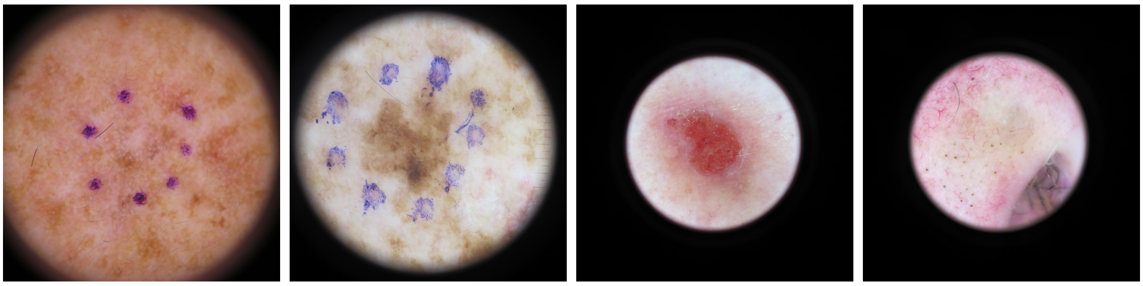
Again, the ViT has best performance, except for the squamous cell carcinoma class where the ResNet has higher recall. Although the ViT's best recall of 93.48% for basal cell carcinoma is quite strong, using a diagnostic model in practice would demand lower false negative rates for all malignant classes. The lower recall for all models for the squamous cell carcinoma diagnosis highlights the challenge of classifying small classes correctly.

6.3 Misclassified examples

To identify any shortcomings with our methods, the test images which were misclassified by all three models were examined. This analysis is inherently ad hoc, but may be useful nonetheless. Of the 1685 images in the test set, 71 were incorrectly classified by all three models. These 71 images were manually inspected, and two key issues were identified:

- Lesion marking: 10 images contained pen marks on the patient's skin to identify the area of the lesion (see (a) and (b) in Figure 10). As our models cannot differentiate between the markings and the lesion, the markings likely contributed to the misclassification.
- Tight cropping: approximately 25 images had very tight cropping around the lesions from the dermoscopy device (see (c) and (d) in Figure 10). All the architectures we used are designed to incorporate context from the whole of the image (through convolution layers in the CNNs and through the transformer encoder in the ViT). Therefore, an overly cropped image may actually reduce the efficacy of these models.

These two issues do not account for all examples that were misclassified and it is natural that some lesions are difficult to diagnose. The problems identified are inherent to the dataset used and would be difficult to solve with model adjustments. However, these issues may be important to consider when constructing dermoscopy datasets in the future.



(a) (b) (c) (d)

Figure 10: Examples of test images incorrectly classified by all three models.

7 Conclusion

In this project, we used Inception, ResNet, and ViT models to classify dermoscopic images. The ViT performed best overall; however, Inception and ResNet showed strong performance with much lower computational cost (under 2 hours of training compared to 20 for the ViT). Overall, our findings highlight the utility of deep learning for skin lesion diagnosis.

All models in our study were trained using the standard cross-entropy loss function. Although this is a widely used for many classification tasks, it treats all classes the same. However, in medical diagnosis, where false negatives for malignant classes are far more serious than false positives, the cross-entropy loss fails to account for the differing risks across classes. Future research should consider other loss functions that explicitly penalize false negatives for malignant classes, such as a focal loss. This alternative training objective could help align model performance with our priorities in the medical setting.

Another significant limitation came from the class distribution in our dataset. Some diagnostic categories, such as dermatofibroma and vascular lesions, are severely underrepresented. This led to uneven classification performance across categories, despite attempting to address the problem with data augmentation, weighted loss functions and balanced class sampling. Future work addressing this issue may require either expanding the dataset (which is often not feasible in clinical settings), or possibly synthetic image generation with GANs or VAEs.

Contributions

- Yung-Cheng (Allan) Chuang: Implemented and evaluated the ViT model. Wrote the code to generate adversarial and attention-noise data. Wrote content for the ViT.
- Declan Fletcher: Wrote code for data cleaning and model testing. Ran ViT experiments. Wrote content on the data, results, and ViT.
- Jack Runchel: Developed code and ran experiments, training and evaluating the Inception model. Wrote content relating to Inception training.
- Ewan Stanich: Implemented and evaluated the ResNet model. Wrote content related to ResNet. Wrote introductory content. Wrote documentation for accessing GPU resources.
- Andrea Luca Taverna: Developed code and ran experiments for the Inception model. Wrote content on Inception model, and conclusion.

Appendices

A Data Augmentation Transform Parameters

Parameters used for PyTorch data transformations.

Transform	Hyperparameters
Random Resized Cropping	Scale range: 0.8–1.0
Random Horizontal Flip	Probability: 0.5
Random Rotation	Angle range: ±15°
Color Jittering	Brightness/Contrast/Saturation: ±0.2; Hue: ±0.1
Normalization	Mean: [0.5, 0.5, 0.5]; Std: [0.5, 0.5, 0.5]

Table 4: Data transformations for Inception v3 with hyperparameters.

Transform	Hyperparameters
Random Resized Cropping	Scale range: 0.8–1.0
Random Horizontal Flip	Probability: 0.5
Random Vertical Flip	Probability: 0.5
Random Rotation	Angle range: ±15°
Color Jittering	Brightness/Contrast/Saturation: ±0.2; Hue: ±0.1
Gaussian Blur	Kernel size: 3
Normalization	Mean: [0.485, 0.456, 0.406]; Std: [0.229, 0.224, 0.225]

Table 5: Data transformations for the WideResNet101 with hyperparameters.

Transform	Hyperparameters
Random Horizontal Flip	Probability: 0.5
Random Rotation	Angle range: $0°, 170°$
Color Jittering	Brightness/Contrast/Saturation: $\pm 0.2, \pm 0.2, \pm 0.2$
Random Affine	translate $0.1, 0.1$
Random Erasing	p=0.5, scale=$0.2, 0.2$, ratio=$0.3, 3.3$
Normalization	Mean: [0.5, 0.5, 0.5]; Std: [0.5, 0.5, 0.5]

Table 6: Data transformations for the ViT with hyperparameters.

B Base Model Loss Curves

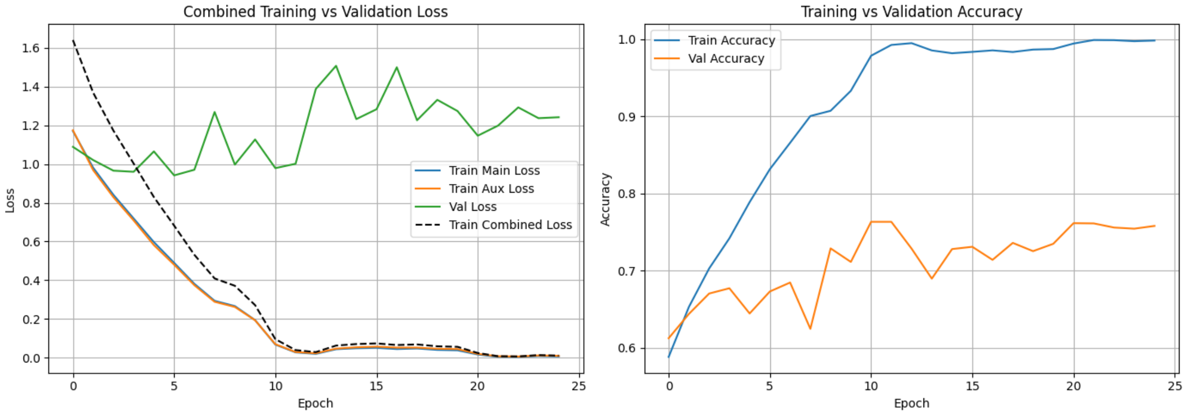


Figure 11: Base Inception model training and validation loss and accuracy.

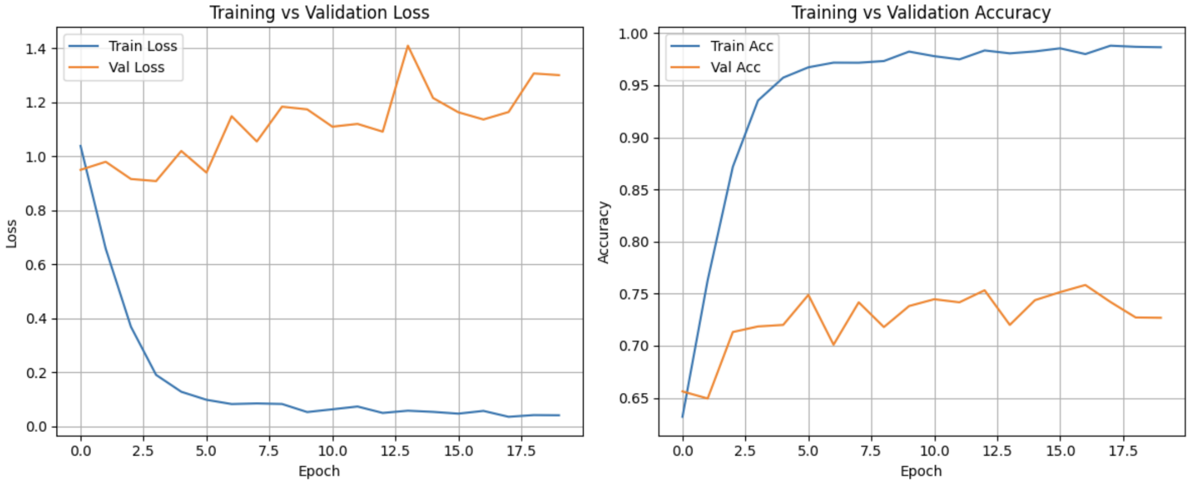


Figure 12: Base ResNet model training and validation loss and accuracy.

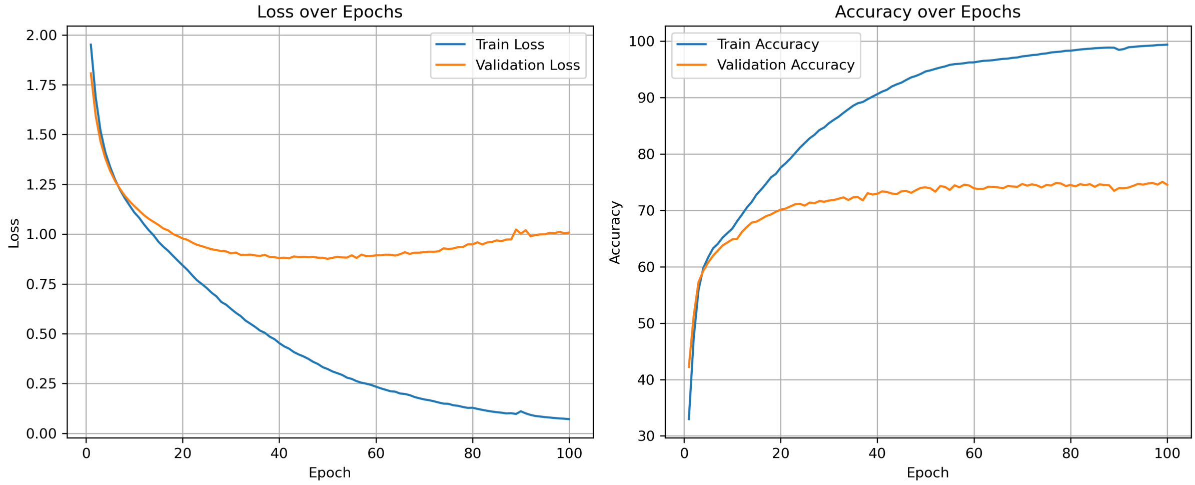


Figure 13: Base ViT model training and validation loss and accuracy.

C Classification Reports

	Precision	Recall	F1-score	Support
Melanoma	0.87	0.87	0.87	400
Nevus	0.87	0.93	0.90	565
Basal cell carcinoma	0.86	0.88	0.87	368
Squamous cell carcinoma	0.74	0.62	0.68	56
Dermatofibroma	0.83	0.59	0.69	17
Benign keratosis	0.85	0.66	0.74	155
Actinic keratosis	0.73	0.76	0.75	109
Vascular lesion	0.79	0.73	0.76	15
Accuracy			0.85	1685
Macro average	0.82	0.76	0.78	1685
Weighted average	0.85	0.85	0.85	1685

Table 7: Classification report on the test set for the final Inception model.

	Precision	Recall	F1-score	Support
Melanoma	0.89	0.88	0.89	400
Nevus	0.90	0.94	0.92	565
Basal cell carcinoma	0.87	0.92	0.90	368
Squamous cell carcinoma	0.77	0.77	0.77	56
Dermatofibroma	1.00	0.59	0.74	17
Benign keratosis	0.84	0.78	0.81	155
Actinic keratosis	0.83	0.73	0.78	109
Vascular lesion	1.00	0.73	0.85	15
Accuracy			0.88	1685
Macro average	0.89	0.79	0.83	1685
Weighted average	0.88	0.88	0.88	1685

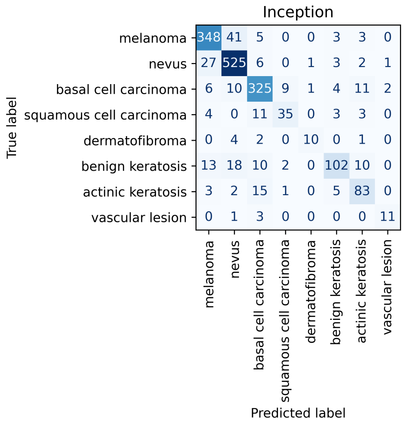
Table 8: Classification report on the test set for the final ResNet model.

	Precision	Recall	F1-score	Support
Melanoma	0.93	0.90	0.92	400
Nevus	0.91	0.95	0.93	565
Basal cell carcinoma	0.86	0.93	0.90	368
Squamous cell carcinoma	0.95	0.66	0.78	56
Dermatofibroma	0.85	0.65	0.73	17
Benign keratosis	0.88	0.78	0.83	155
Actinic keratosis	0.83	0.81	0.82	109
Vascular lesion	1.00	0.80	0.89	15
Accuracy			0.90	1685
Macro average	0.90	0.81	0.85	1685
Weighted average	0.90	0.90	0.90	1685

Table 9: Classification report on the test set for the final ViT model.

D Confusion matrices

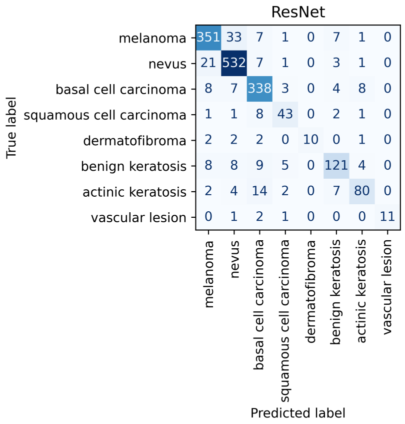
Below are the confusion matrices for each models predictions on the test set.



Inception

True label \ Predicted label	melanoma	nevus	basal cell carcinoma	squamous cell carcinoma	dermatofibroma	benign keratosis	actinic keratosis	vascular lesion
melanoma	348	41	5	0	0	3	3	0
nevus	27	525	6	0	1	3	2	1
basal cell carcinoma	6	10	325	9	1	4	11	2
squamous cell carcinoma	4	0	11	35	0	3	3	0
dermatofibroma	0	4	2	0	10	0	1	0
benign keratosis	13	18	10	2	0	102	10	0
actinic keratosis	3	2	15	1	0	5	83	0
vascular lesion	0	1	3	0	0	0	0	11

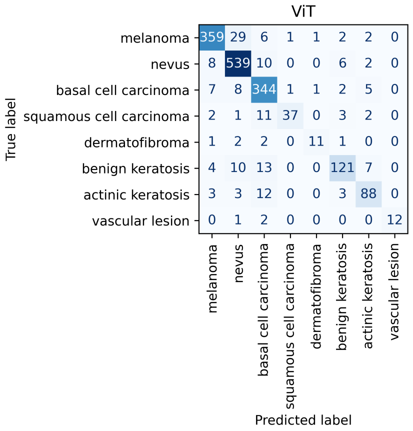
Figure 14: Inception confusion matrix.



ResNet

True label \ Predicted label	melanoma	nevus	basal cell carcinoma	squamous cell carcinoma	dermatofibroma	benign keratosis	actinic keratosis	vascular lesion
melanoma	351	33	7	1	0	7	1	0
nevus	21	532	7	1	0	3	1	0
basal cell carcinoma	8	7	338	3	0	4	8	0
squamous cell carcinoma	1	1	8	43	0	2	1	0
dermatofibroma	2	2	2	0	10	0	1	0
benign keratosis	8	8	9	5	0	121	4	0
actinic keratosis	2	4	14	2	0	7	80	0
vascular lesion	0	1	2	1	0	0	0	11

Figure 15: ResNet confusion matrix.



ViT

True label \ Predicted label	melanoma	nevus	basal cell carcinoma	squamous cell carcinoma	dermatofibroma	benign keratosis	actinic keratosis	vascular lesion
melanoma	359	29	6	1	1	2	2	0
nevus	8	539	10	0	0	6	2	0
basal cell carcinoma	7	8	344	1	1	2	5	0
squamous cell carcinoma	2	1	11	37	0	3	2	0
dermatofibroma	1	2	2	0	11	1	0	0
benign keratosis	4	10	13	0	0	121	7	0
actinic keratosis	3	3	12	0	0	3	88	0
vascular lesion	0	1	2	0	0	0	0	12

Figure 16: ViT confusion matrix.

References

Brinker, T. J. et al. (2021). "Deep Neural Networks Are Superior to Dermatologists in Melanoma Image Classification: An International Multicenter Performance Evaluation". In: *Journal of the American Academy of Dermatology* 86.1, pp. 49–55.

Chen, L. et al. (2021). "Rethinking Image Classification with Vision Transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10102–10112.

Dosovitskiy, A. et al. (2020). "An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale". In: *arXiv preprint arXiv:2010.11929*.

— (2021). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.* arXiv: `2010.11929 [cs.CV]`. URL: `https://arxiv.org/abs/2010.11929`.

Esteva, A. et al. (2017). "Dermatologist-level classification of skin cancer with deep neural networks". In: *Nature* 542.7639, pp. 115–118. DOI: `10.1038/nature21056`. URL: `https://www.nature.com/articles/nature21056`.

He, K. et al. (2015). "Deep Residual Learning for Image Recognition". In: *arXiv preprint arXiv:1512.03385*. arXiv: `1512.03385 [cs.CV]`. URL: `https://arxiv.org/abs/1512.03385`.

Hernández-Pérez, C. et al. (June 2024). "BCN20000: Dermoscopic Lesions in the Wild". In: *Scientific Data* 11.1. ISSN: 2052-4463. DOI: `10.1038/s41597-024-03387-w`. URL: `http://dx.doi.org/10.1038/s41597-024-03387-w`.

Hugging Face (2025). *Vision Transformer (ViT).* `https://huggingface.co/docs/transformers/model_doc/vit#vision-transformer-vit`. Accessed: 2025-06-08.

Inception v3 - PyTorch Hub (2019). `https://pytorch.org/hub/pytorch_vision_inception_v3/`. Accessed: 2025-04-01.

ISIC (2025). *ISIC Archive API.* `https://api.isic-archive.com/api/docs/swagger/`. Accessed: April 2025.

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems.* Vol. 25, pp. 1097–1105.

Liu, X. (2023). *The architecture of the ResNet-101 model.* `https://www.researchgate.net/figure/The-architecture-of-the-ResNet-101-model_fig7_360641512`. Accessed: 2025-05-30.

Pan, Y. et al. (2023). "Fundus image classification using Inception V3 and ResNet-50 for the early diagnostics of fundus diseases". In: *Frontiers in Physiology* 14. ISSN: 1664-042X. DOI: `10.3389/fphys.2023.1126780`. URL: `https://www.frontiersin.org/journals/physiology/articles/10.3389/fphys.2023.1126780/full`.

Park, S. et al. (2022). "Multi-task vision transformer using low-level chest X-ray feature corpus for COVID-19 diagnosis and severity quantification". In: *Medical Image Analysis* 75, p. 102299. ISSN: 1361-8415. DOI: `https://doi.org/10.1016/j.media.2021.102299`. URL: `https://www.sciencedirect.com/science/article/pii/S1361841521003443`.

Szegedy, C. et al. (2014). "Going deeper with convolutions". In: *arXiv preprint arXiv:1409.4842*.

Szegedy, C. et al. (2016). "Rethinking the Inception Architecture for Computer Vision". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826.

Torchvision Contributors (2025). *torchvision.models.wide_resnet101_2.* `https://pytorch.org/vision/stable/models/generated/torchvision.models.wide_resnet101_2.html`. Accessed: 2025-05-30.

Vaswani, A. et al. (2023). *Attention Is All You Need.* arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

Wu, Y. et al. (2022). "Skin Cancer Classification With Deep Learning: A Systematic Review". In: *Frontiers in Oncology* Volume 12 - 2022. ISSN: 2234-943X. DOI: 10.3389/fonc.2022.893972. URL: https://www.frontiersin.org/journals/oncology/articles/10.3389/fonc.2022.893972.

Zagoruyko, S. and N. Komodakis (2016). "Wide Residual Networks". In: *arXiv preprint arXiv:1605.07146.* arXiv: 1605.07146 [cs.CV]. URL: https://arxiv.org/abs/1605.07146.

We give consent for this to be used as a teaching resource.